LLMPirate: LLMs for Black-box Hardware IP Piracy

Vasudev Gohil

Matthew DeLorenzo

Veera Vishwa Achuta

Sai Venkat Nallam

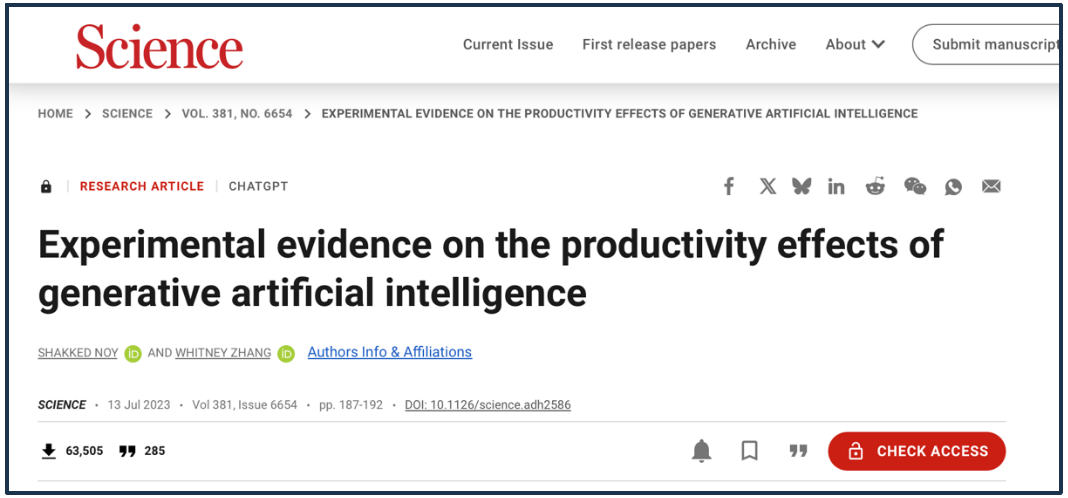
Joey See

Jeyavijayan Rajendran

TEXAS A&M UNIVERSITY Engineering

SIEMENS

LLMs are Everywhere



Science

Current Issue First release papers Archive About Submit manuscript

HOME > SCIENCE > VOL. 381, NO. 6654 > EXPERIMENTAL EVIDENCE ON THE PRODUCTIVITY EFFECTS OF GENERATIVE ARTIFICIAL INTELLIGENCE

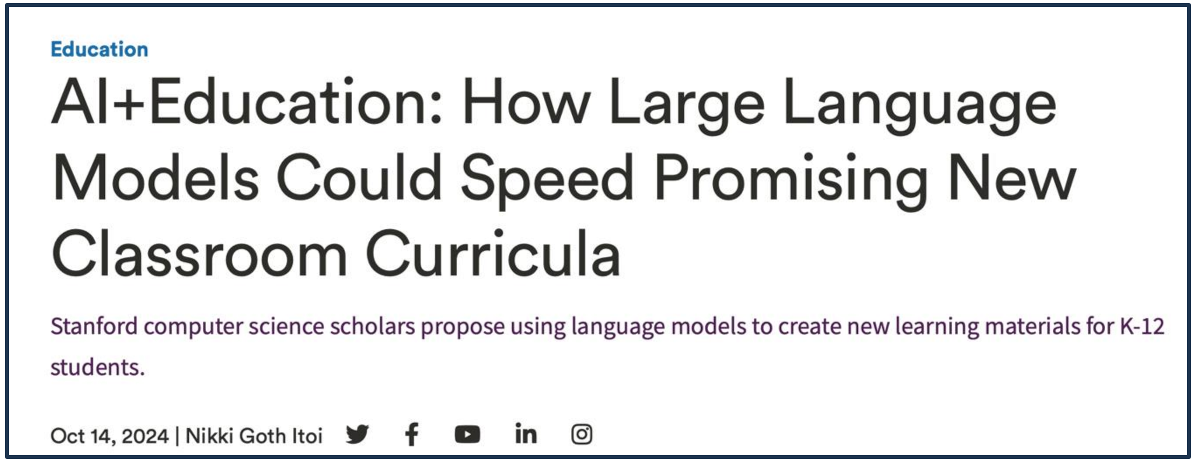
RESEARCH ARTICLE | CHATGPT

Experimental evidence on the productivity effects of generative artificial intelligence

SHAKKED NOY AND WHITNEY ZHANG Authors Info & Affiliations

SCIENCE • 13 Jul 2023 • Vol 381, Issue 6654 • pp. 187-192 • DOI: 10.1126/science.adh2586

63,505 285 CHECK ACCESS

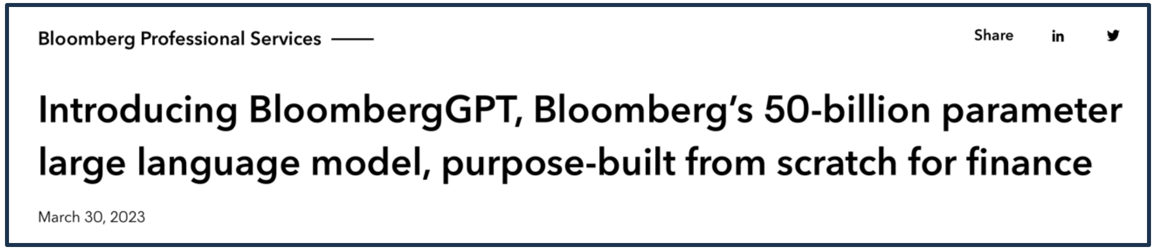


Education

AI+Education: How Large Language Models Could Speed Promising New Classroom Curricula

Stanford computer science scholars propose using language models to create new learning materials for K-12 students.

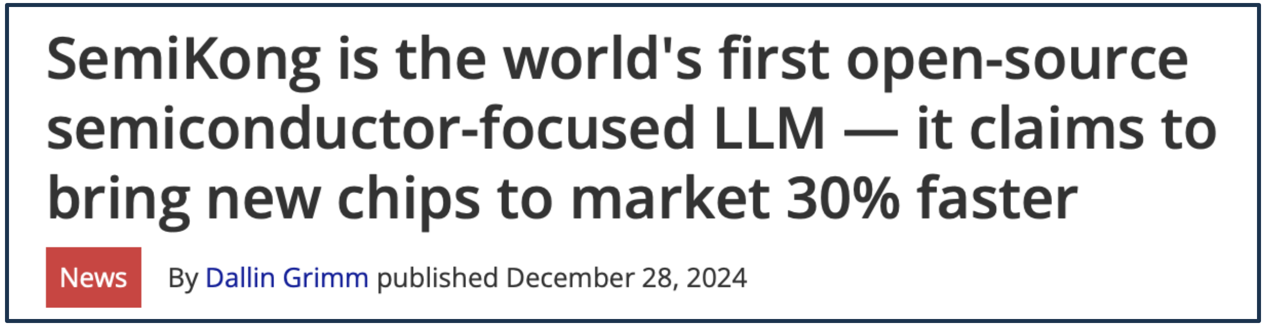
Oct 14, 2024 | Nikki Goth Itoi



Bloomberg Professional Services —— Share

Introducing BloombergGPT, Bloomberg's 50-billion parameter large language model, purpose-built from scratch for finance

March 30, 2023



SemiKong is the world's first open-source semiconductor-focused LLM — it claims to bring new chips to market 30% faster

News By Dallin Grimm published December 28, 2024

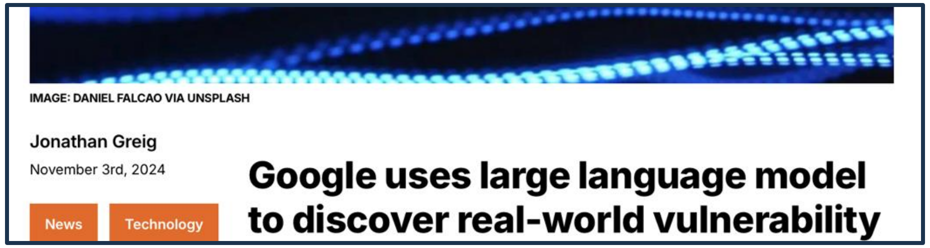


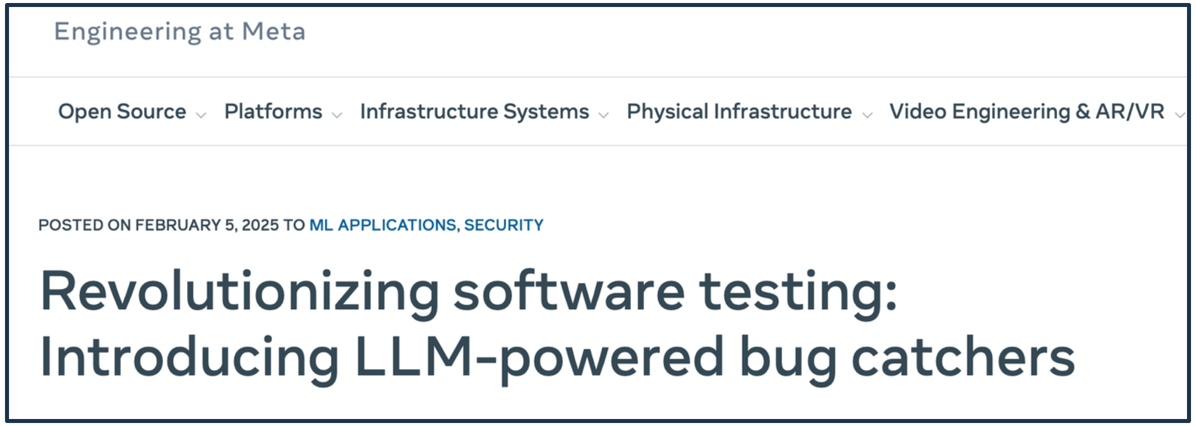
IMAGE: DANIEL FALCAO VIA UNSPLASH

Jonathan Greig

November 3rd, 2024

News Technology

Google uses large language model to discover real-world vulnerability



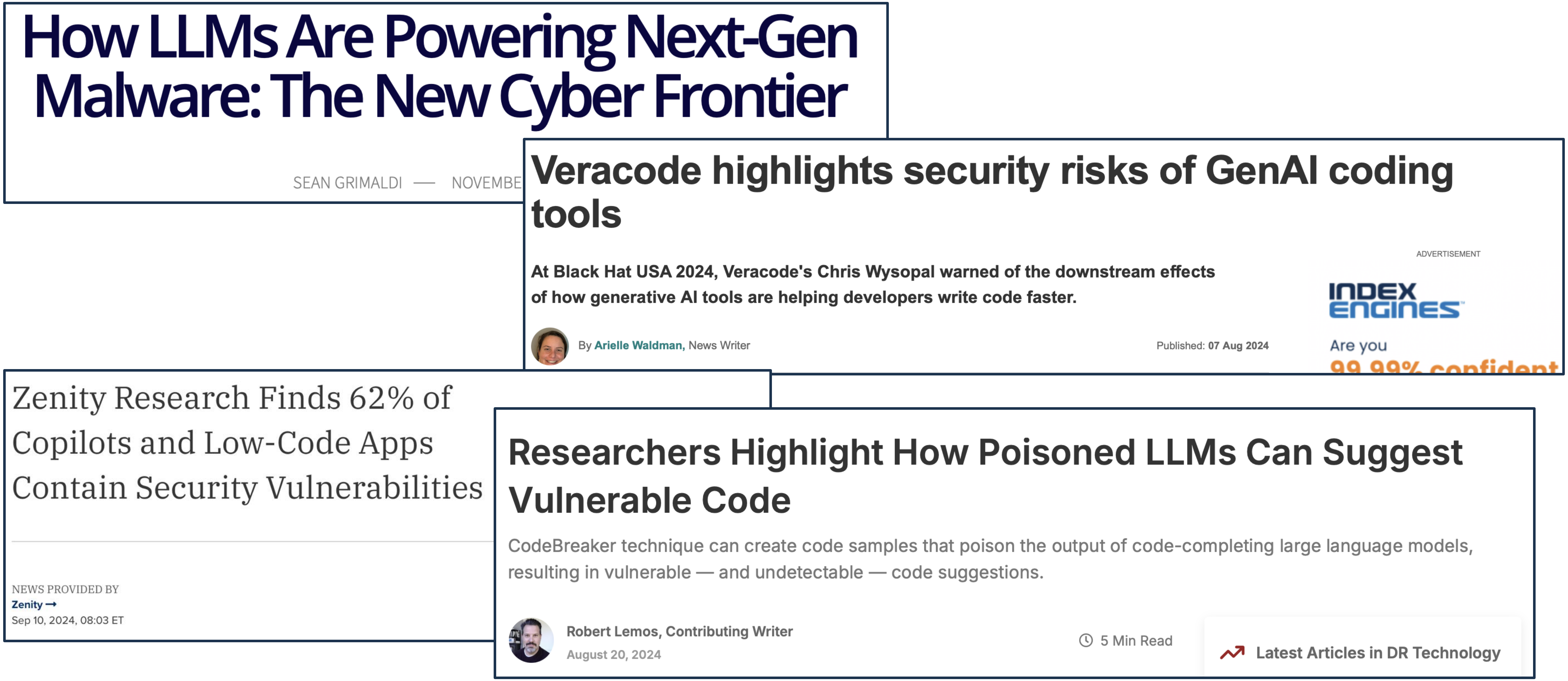
Engineering at Meta

Open Source Platforms Infrastructure Systems Physical Infrastructure Video Engineering & AR/VR

POSTED ON FEBRUARY 5, 2025 TO ML APPLICATIONS, SECURITY

Revolutionizing software testing: Introducing LLM-powered bug catchers

LLM-related Attack Vectors



How LLMs Are Powering Next-Gen Malware: The New Cyber Frontier

SEAN GRIMALDI — NOVEMBE

Veracode highlights security risks of GenAI coding tools

At Black Hat USA 2024, Veracode's Chris Wysopal warned of the downstream effects of how generative AI tools are helping developers write code faster.

By Arielle Waldman, News Writer

Published: 07 Aug 2024

Zenity Research Finds 62% of Copilots and Low-Code Apps Contain Security Vulnerabilities

NEWS PROVIDED BY
Zenity →
Sep 10, 2024, 08:03 ET

Researchers Highlight How Poisoned LLMs Can Suggest Vulnerable Code

CodeBreaker technique can create code samples that poison the output of code-completing large language models, resulting in vulnerable — and undetectable — code suggestions.

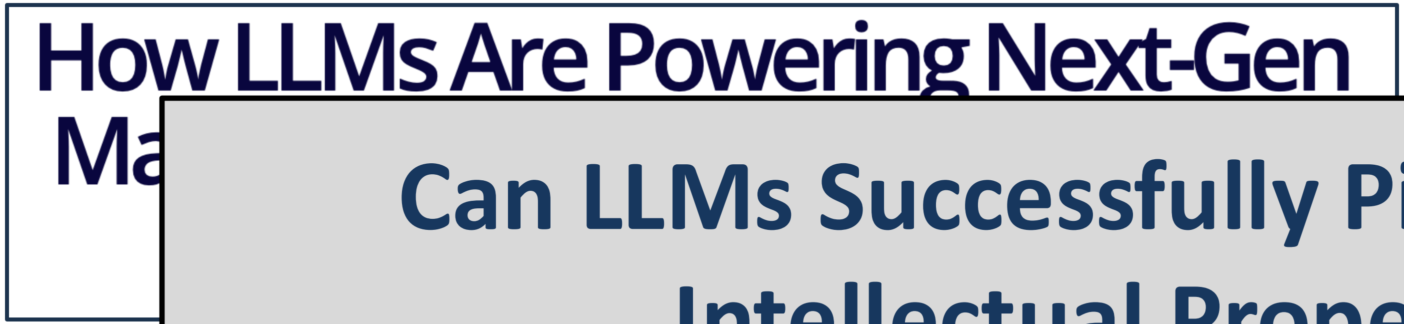
Robert Lemos, Contributing Writer
August 20, 2024

5 Min Read

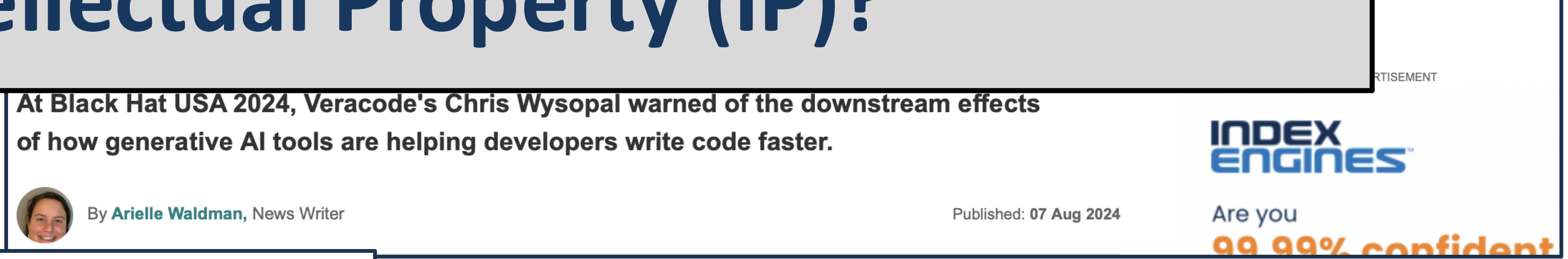
Latest Articles in DR Technology

LLM-related Attack Vectors

Can LLMs Successfully Pirate Hardware Intellectual Property (IP)?



How LLMs Are Powering Next-Gen Ma

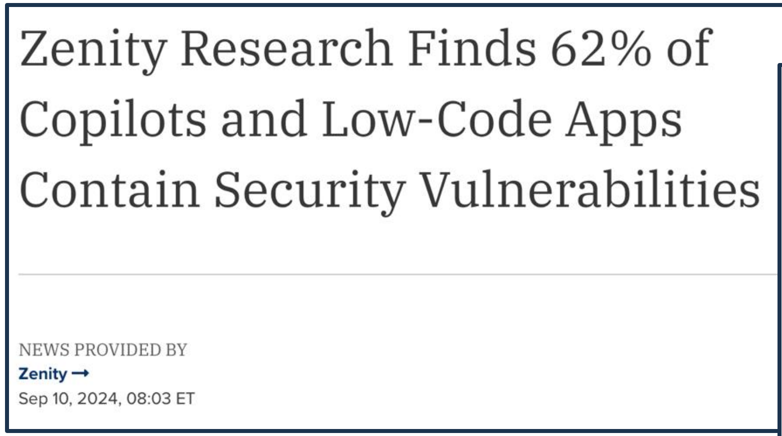


g

At Black Hat USA 2024, Veracode's Chris Wysopal warned of the downstream effects of how generative AI tools are helping developers write code faster.

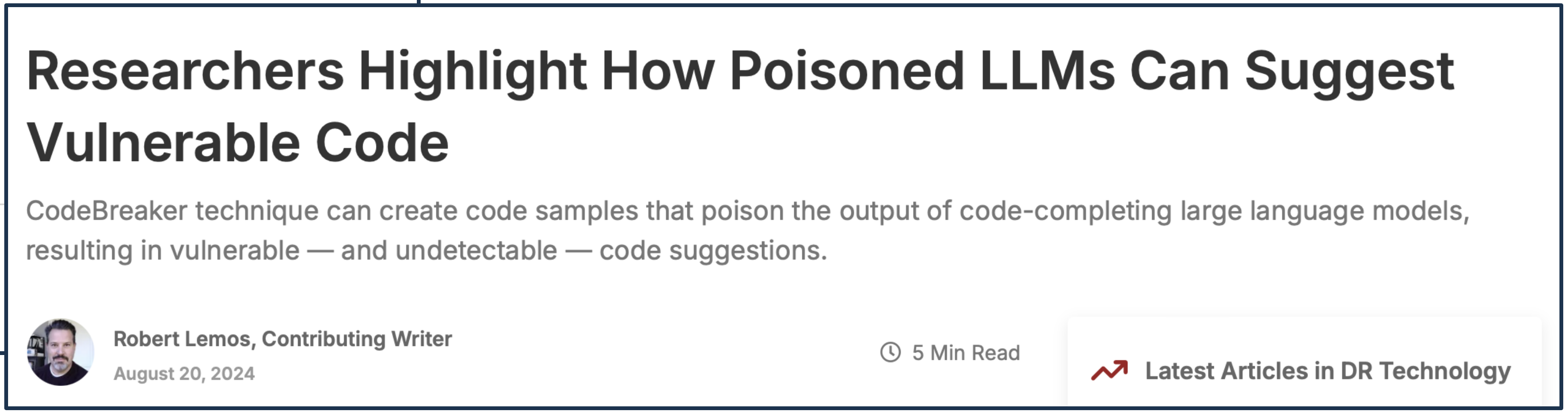
By Arielle Waldman, News Writer

Published: 07 Aug 2024



Zenity Research Finds 62% of Copilots and Low-Code Apps Contain Security Vulnerabilities

NEWS PROVIDED BY
Zenity →
Sep 10, 2024, 08:03 ET



Researchers Highlight How Poisoned LLMs Can Suggest Vulnerable Code

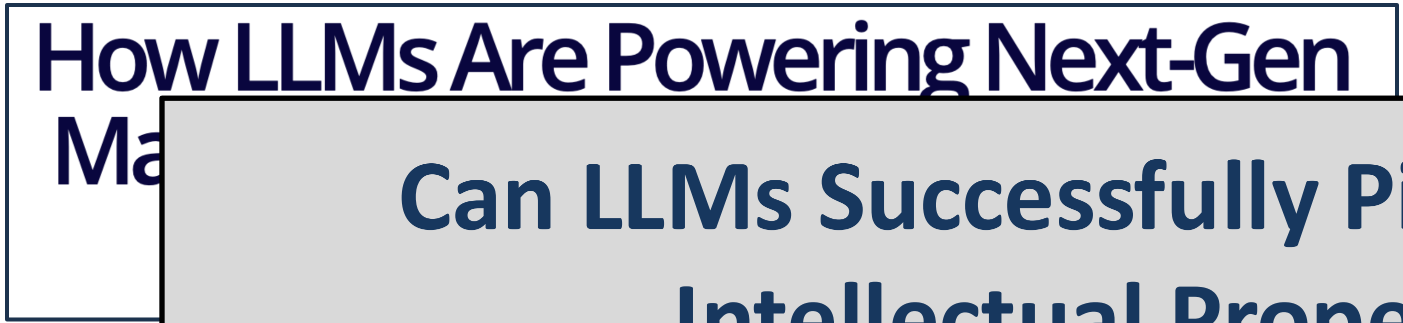
CodeBreaker technique can create code samples that poison the output of code-completing large language models, resulting in vulnerable — and undetectable — code suggestions.

Robert Lemos, Contributing Writer
August 20, 2024

5 Min Read

Latest Articles in DR Technology

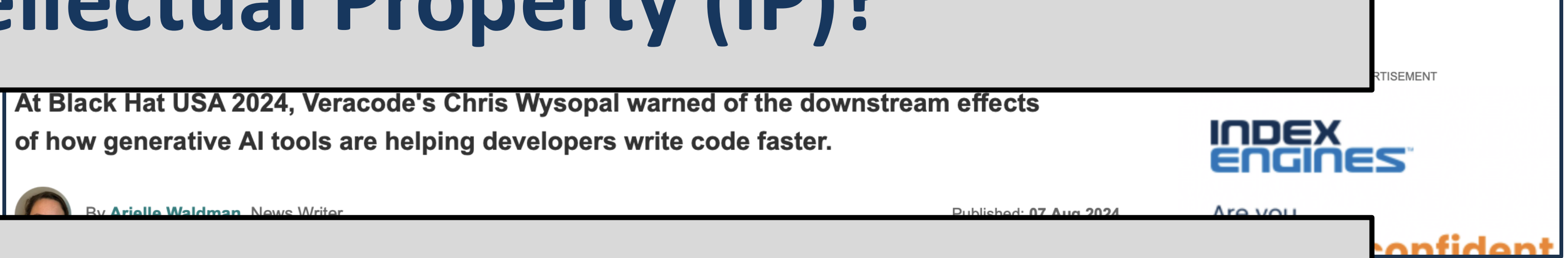
LLM-related Attack Vectors



How LLMs Are Powering Next-Gen

Ma

Can LLMs Successfully Pirate Hardware Intellectual Property (IP)?



g

RTISEMENT

At Black Hat USA 2024, Veracode's Chris Wysopal warned of the downstream effects of how generative AI tools are helping developers write code faster.

By Arielle Waldman, News Writer

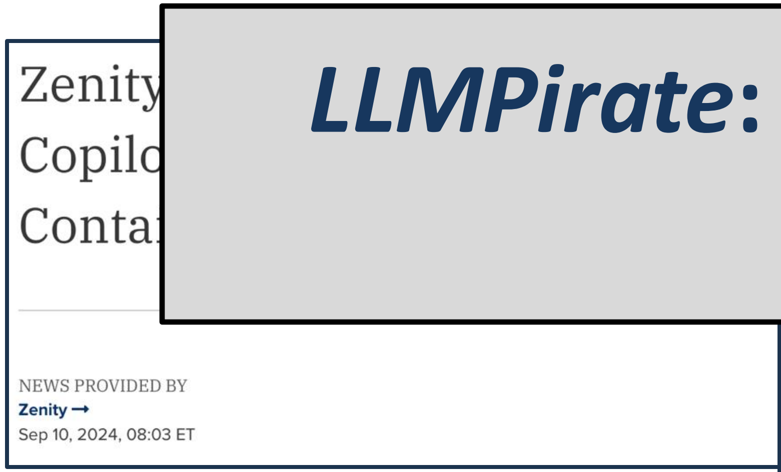
Published: 07 Aug 2024

INDEX ENGINES

Are you

confident

LLMPirate: LLMs for Black-box Hardware IP Piracy



Zenity

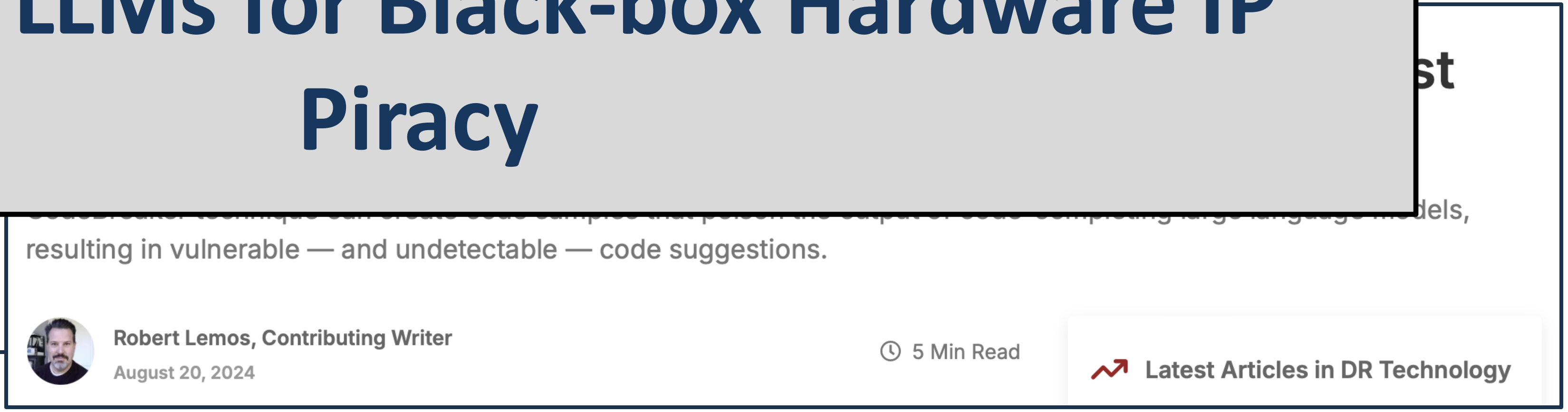
Copilo

Conta

NEWS PROVIDED BY

Zenity →

Sep 10, 2024, 08:03 ET



st

dels, resulting in vulnerable — and undetectable — code suggestions.

Robert Lemos, Contributing Writer

August 20, 2024

5 Min Read

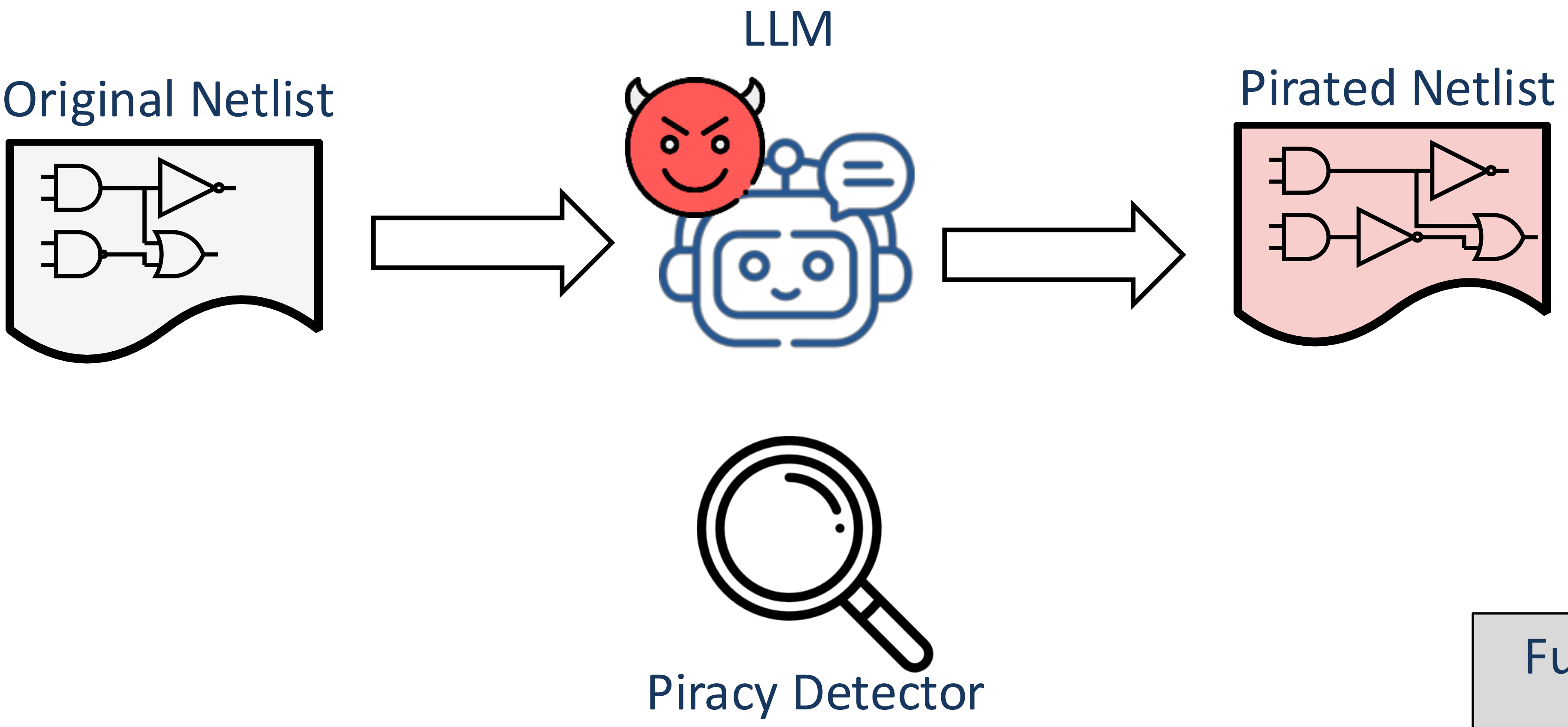
Latest Articles in DR Technology

Threat Model

Piracy Detector

No Modifications

Threat Model



LLM

Original Netlist

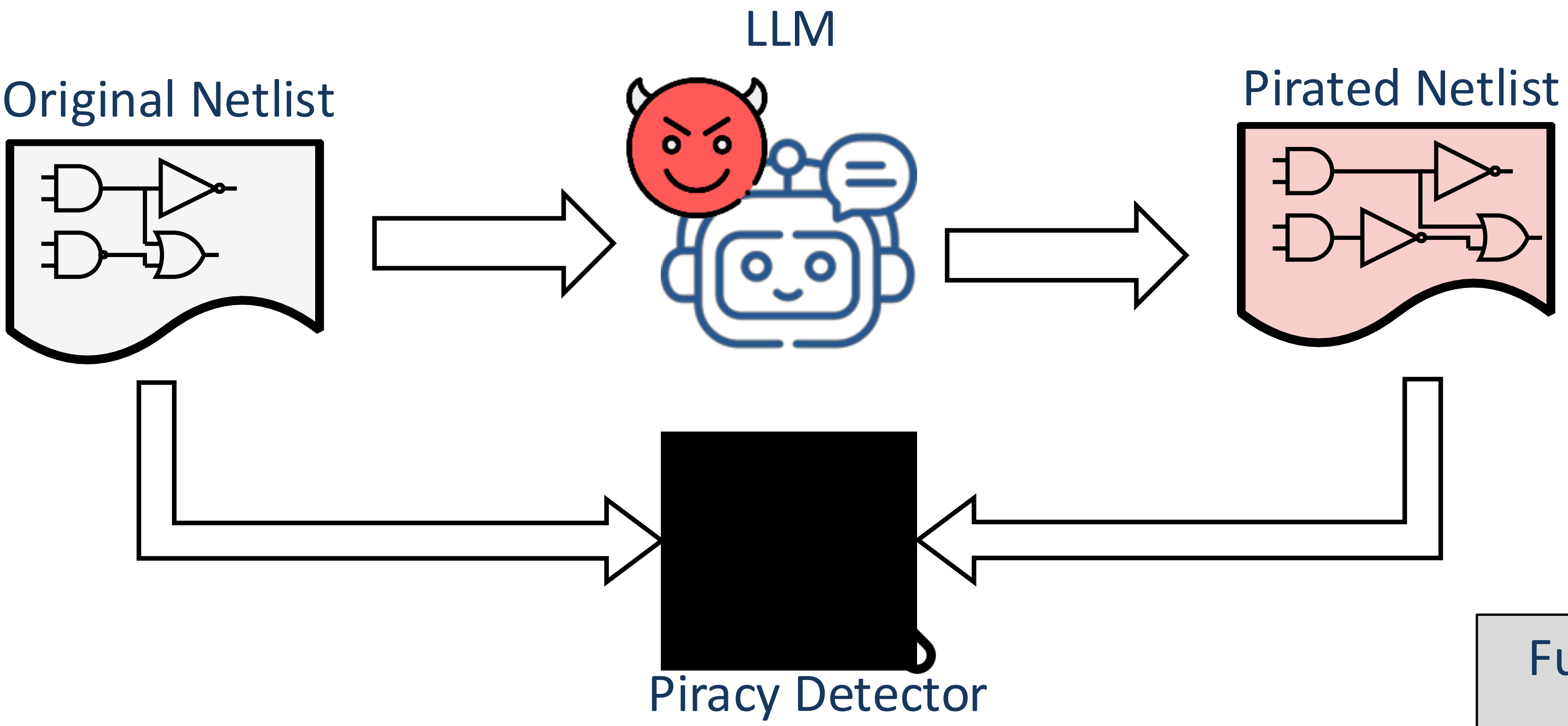
Pirated Netlist

Piracy Detector

No Modifications

Functionally-equivalent Transformations

Threat Model

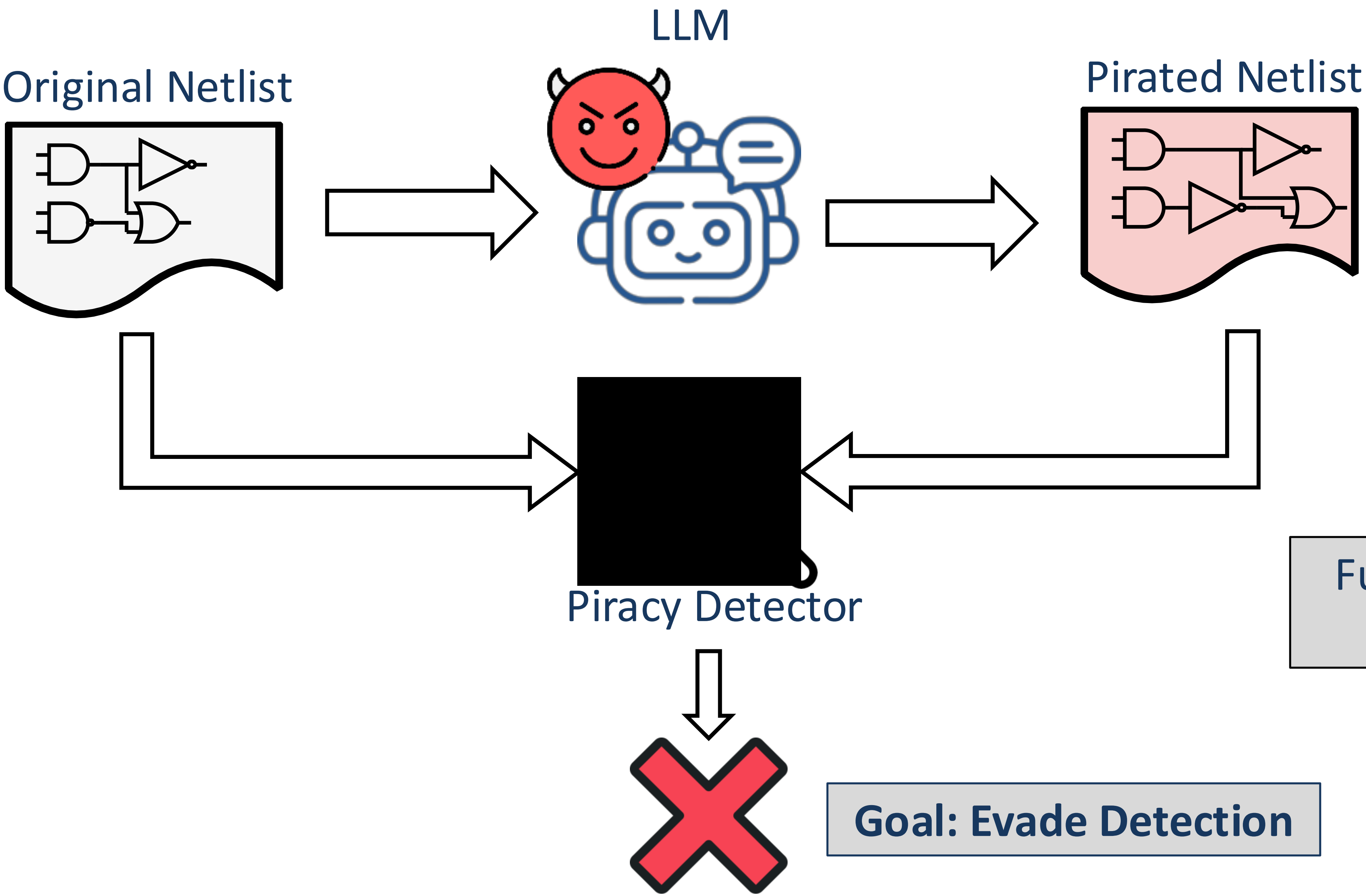


No Modifications

Functionally-equivalent Transformations

Black-box Access

Threat Model



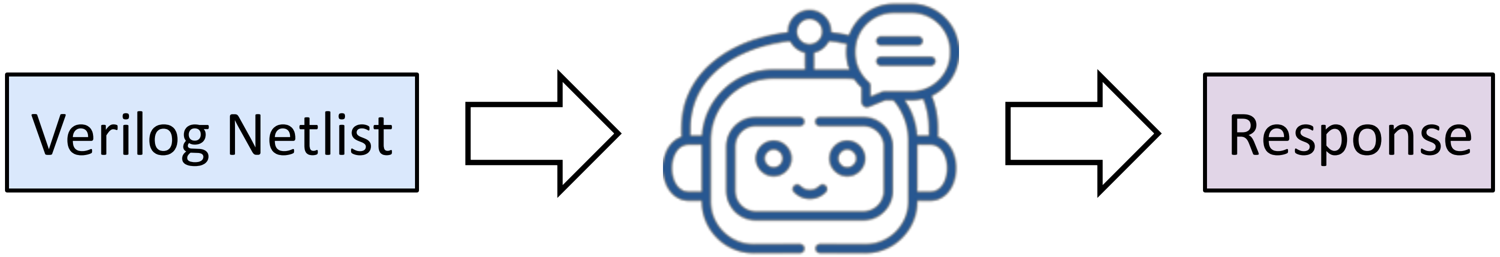
Goal: Evade Detection

No Modifications

Functionally-equivalent Transformations

Black-box Access

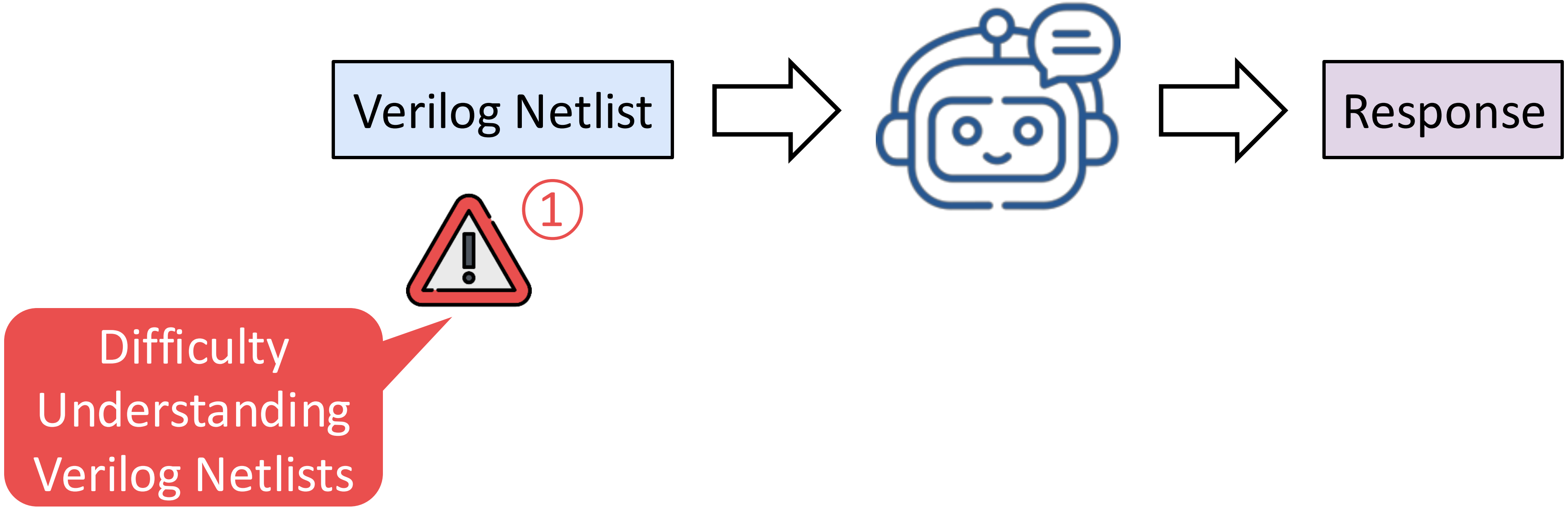
LLMPirate – Preliminary Formulation



Verilog Netlist

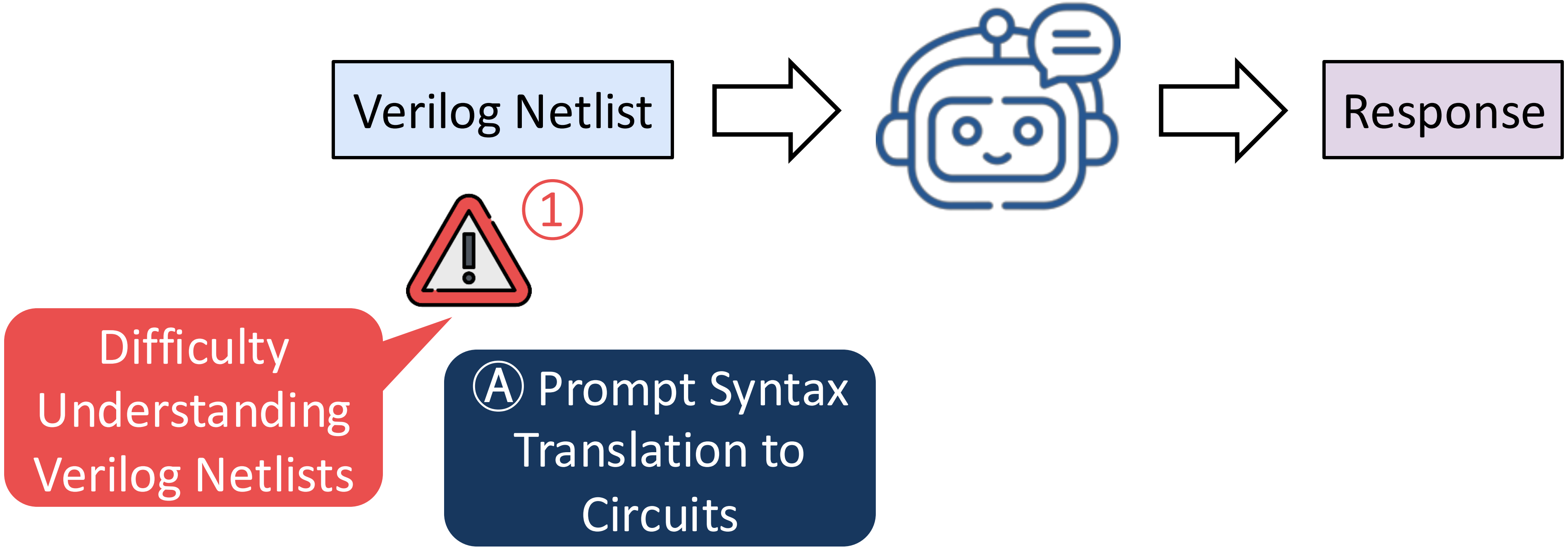
Response

LLMPirate – Challenges and Solutions



Difficulty Understanding Verilog Netlists

LLMPirate – Challenges and Solutions



Verilog Netlist

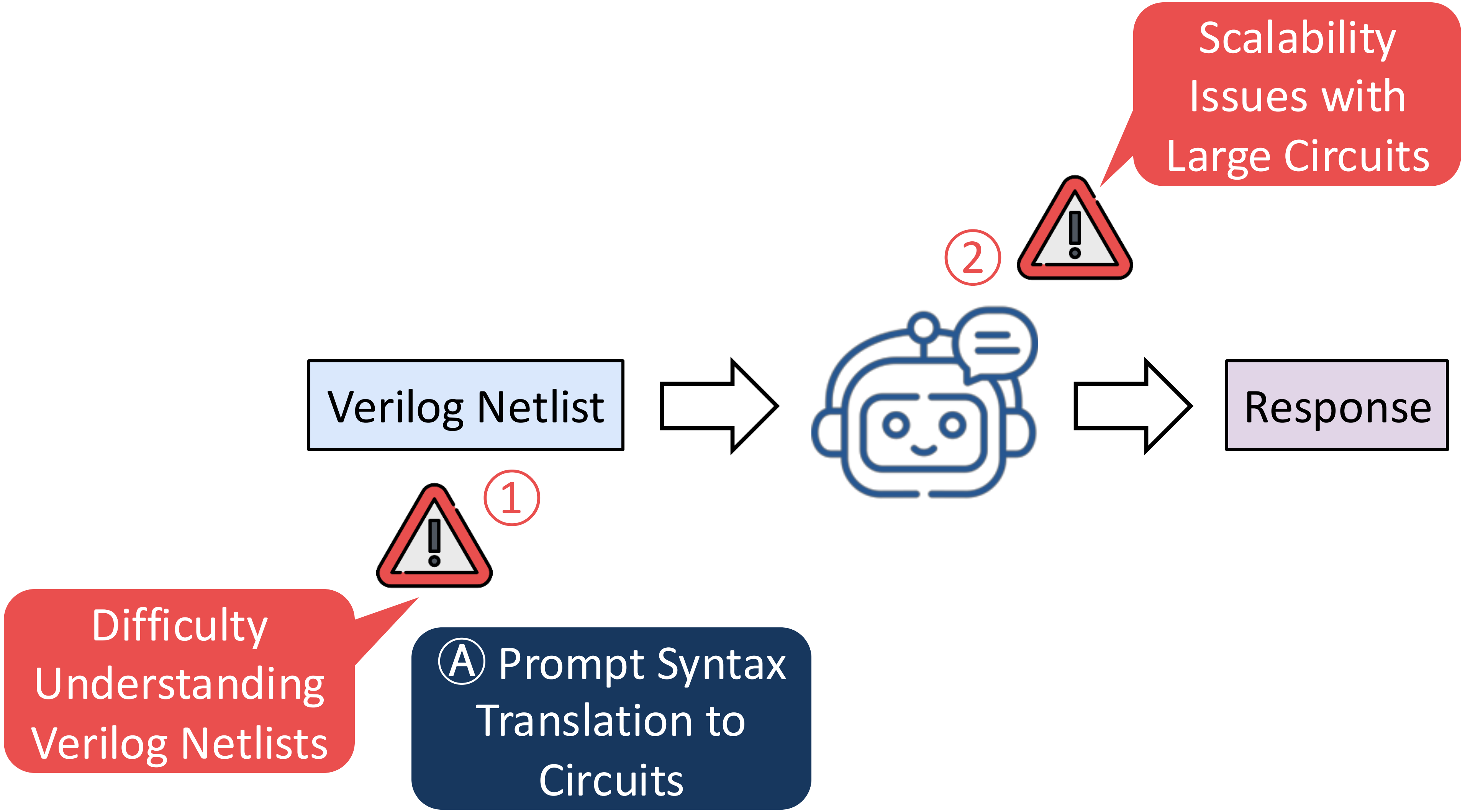
Response

①

Difficulty Understanding Verilog Netlists

Ⓐ Prompt Syntax Translation to Circuits

LLMPirate – Challenges and Solutions



Verilog Netlist

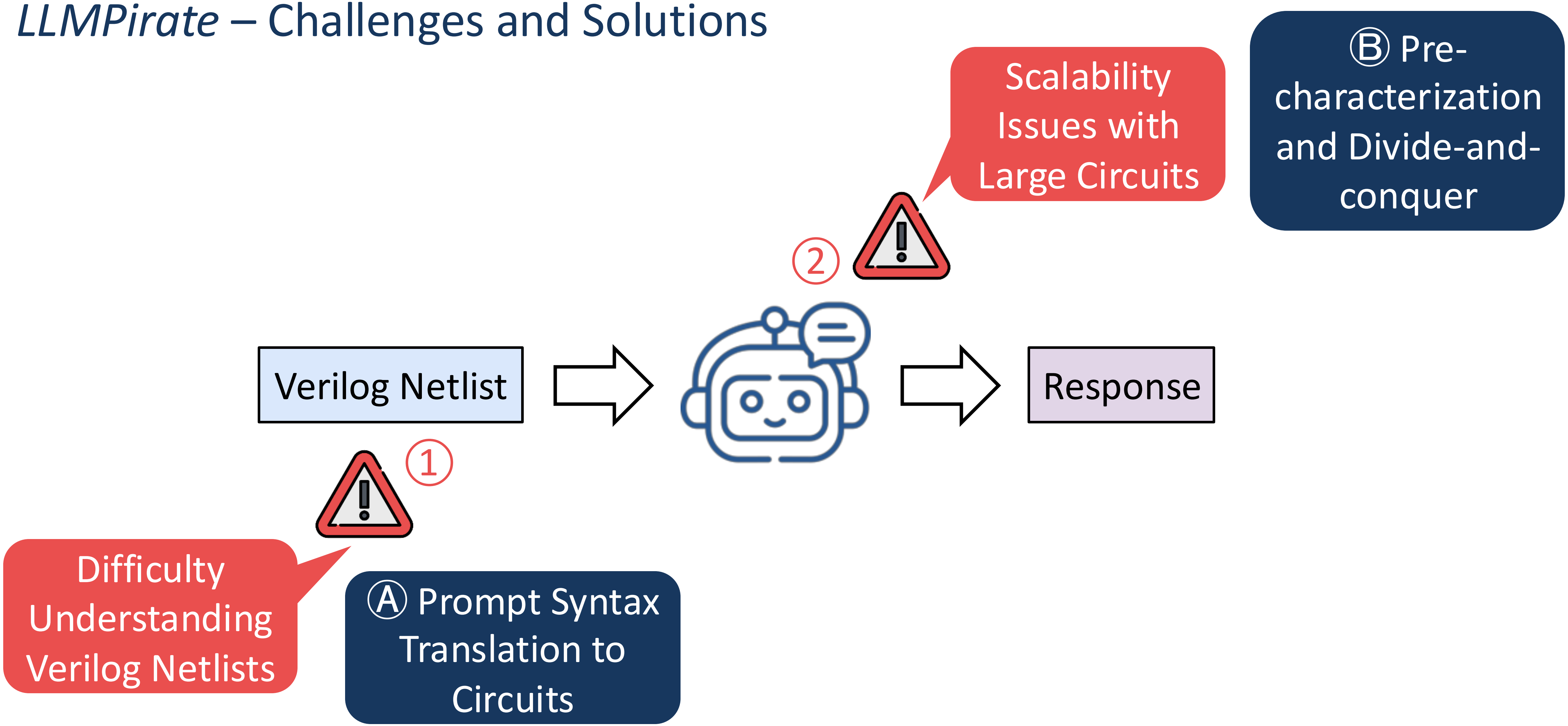
Response

① Difficulty Understanding Verilog Netlists

② Scalability Issues with Large Circuits

Ⓐ Prompt Syntax Translation to Circuits

LLMPirate – Challenges and Solutions



Verilog Netlist

Response

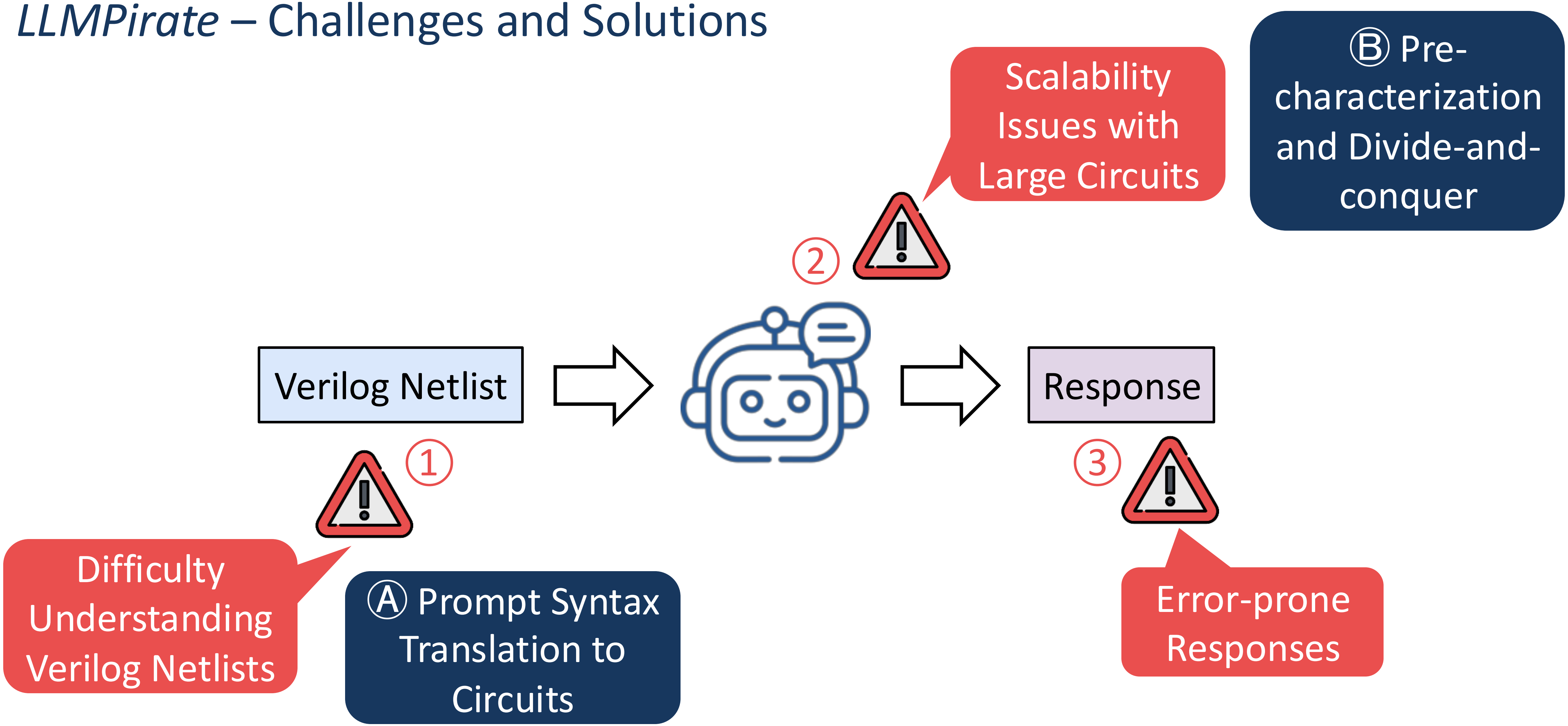
① Difficulty Understanding Verilog Netlists

Ⓐ Prompt Syntax Translation to Circuits

② Scalability Issues with Large Circuits

Ⓑ Pre-characterization and Divide-and-conquer

LLMPirate – Challenges and Solutions



Verilog Netlist

Response

① Difficulty Understanding Verilog Netlists

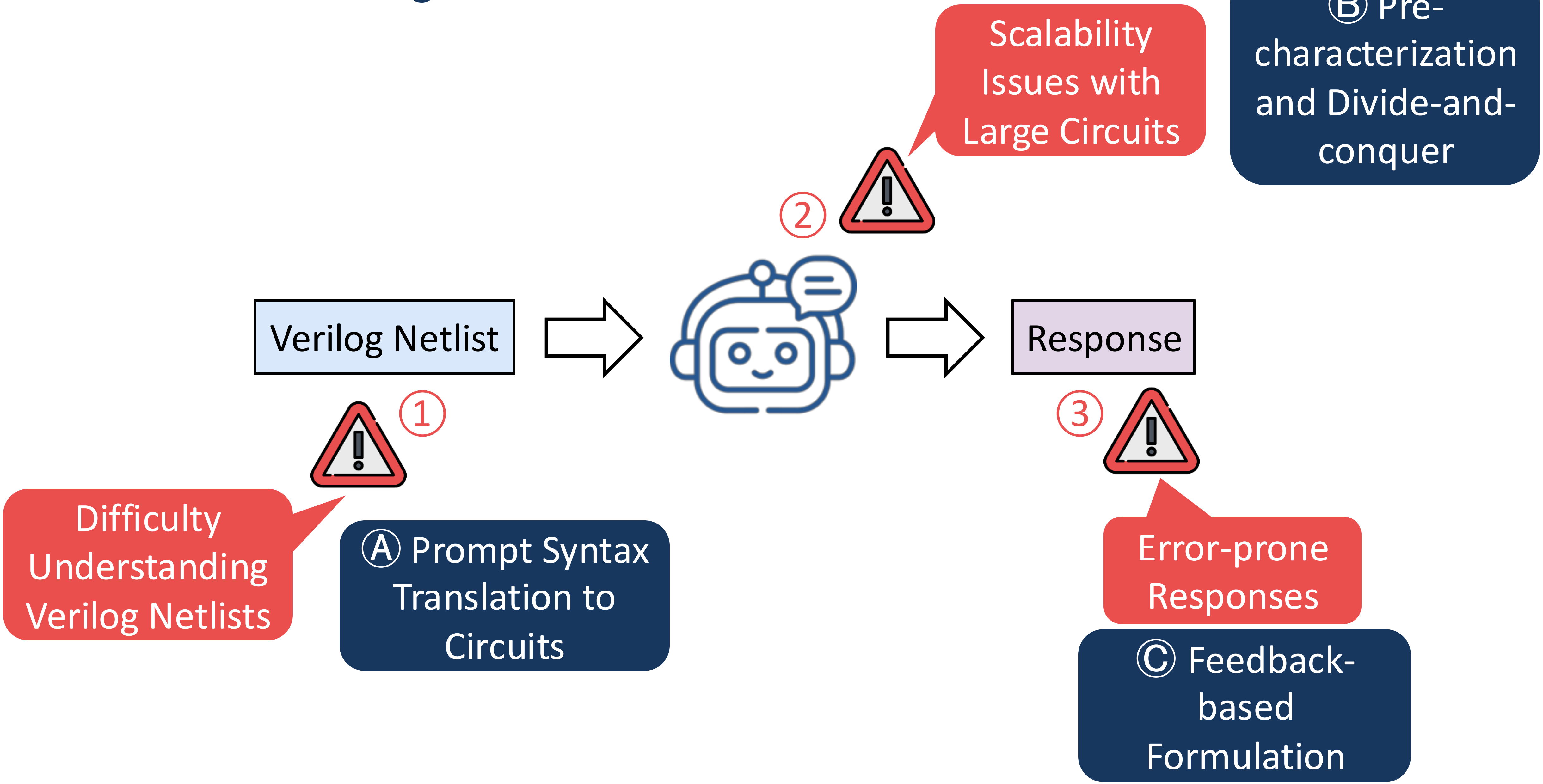
Ⓐ Prompt Syntax Translation to Circuits

② Scalability Issues with Large Circuits

Ⓑ Pre-characterization and Divide-and-conquer

③ Error-prone Responses

LLMPirate – Challenges and Solutions



Verilog Netlist

Response

①

Difficulty Understanding Verilog Netlists

Ⓐ Prompt Syntax Translation to Circuits

②

Scalability Issues with Large Circuits

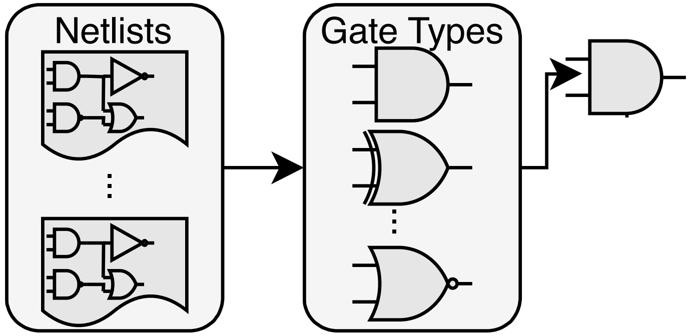
Ⓑ Pre-characterization and Divide-and-conquer

③

Error-prone Responses

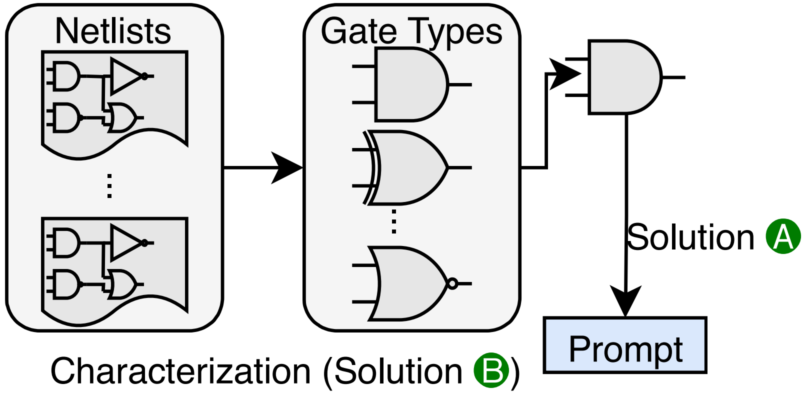
Ⓒ Feedback-based Formulation

LLMPirate – Putting It All Together

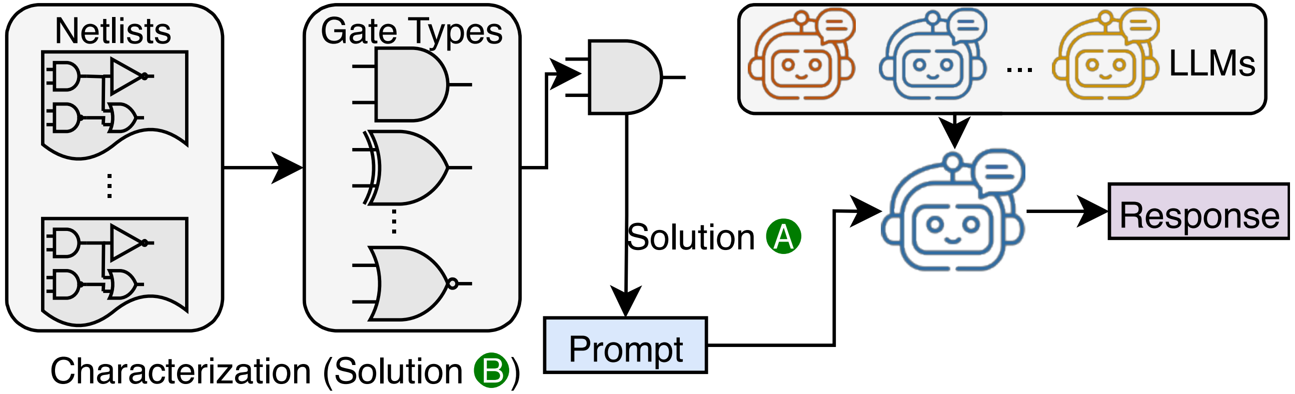


Characterization (Solution Ⓑ)

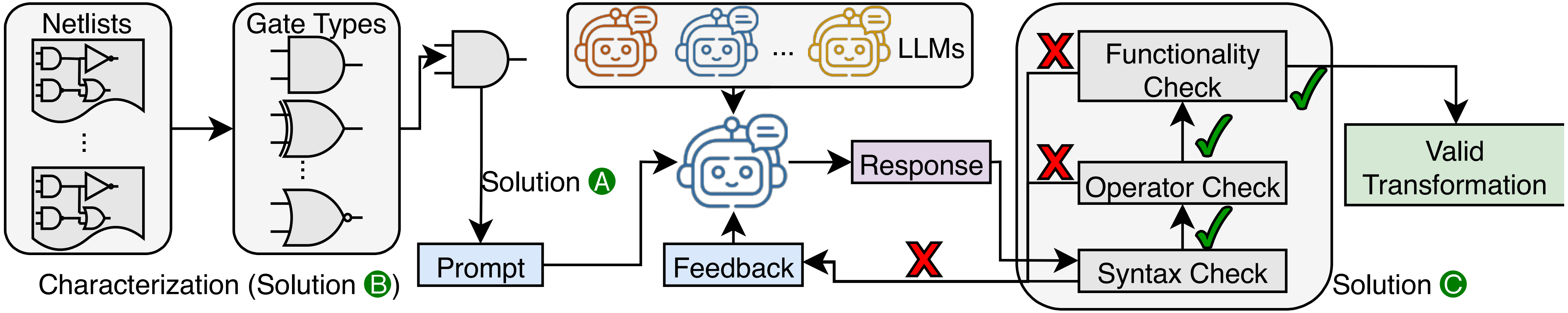
LLMPirate – Putting It All Together



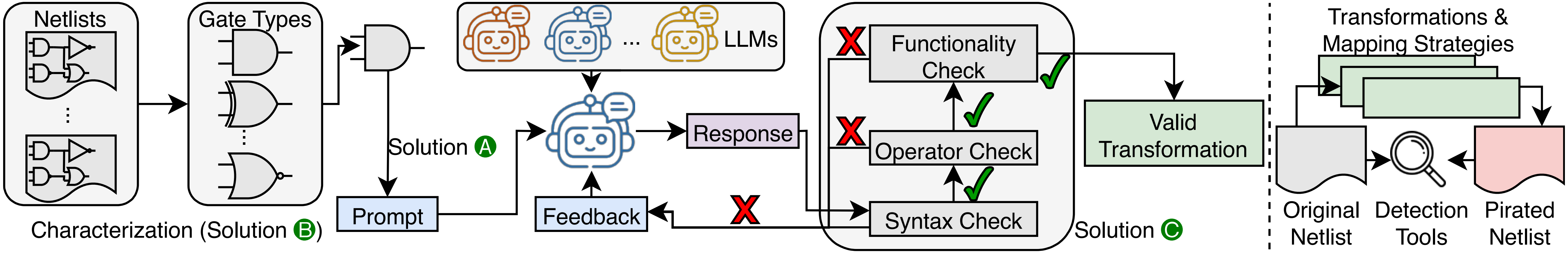
LLMPirate – Putting It All Together



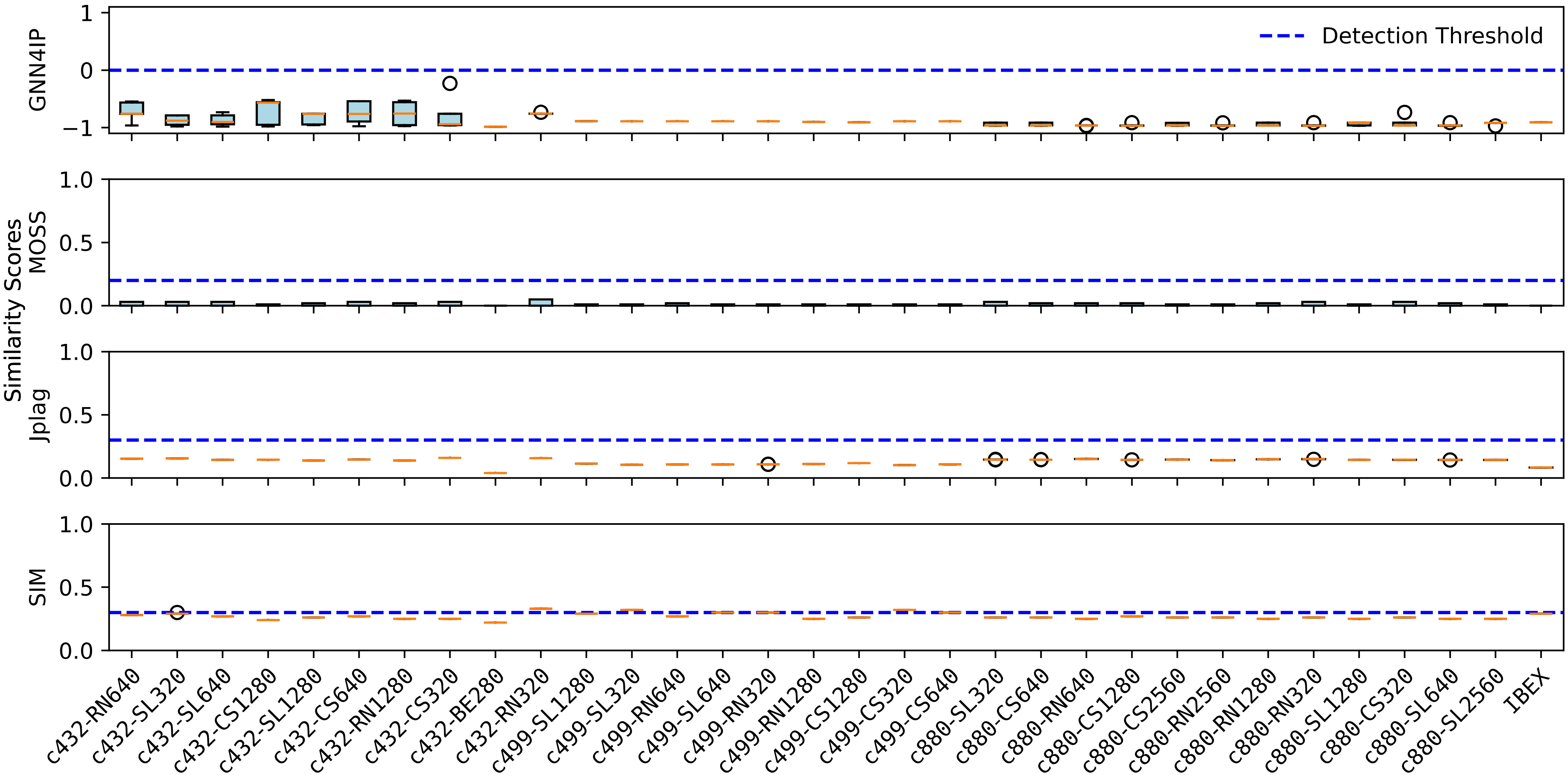
LLMPirate – Putting It All Together



LLMPirate – Putting It All Together



Main Results



LLMPirate Successfully Evades **All** Detection Tools

Ablation Study

# Successes (Sim. Score)	**GNN4IP** [1]	**MOSS** [2]	**Jplag** [3]	**SIM** [4]
LLMPirate\ Solution Ⓐ	0 (N/A)	0 (N/A)	0 (N/A)	0 (N/A)
LLMPirate\ Solution Ⓑ	0 (N/A)	0 (N/A)	0 (N/A)	0 (N/A)
LLMPirate\ Solution Ⓒ	32 (-0.75)	32 (0.01)	32 (0.20)	7 (0.32)
LLMPirate	32 (-0.88)	32 (0.01)	32 (0.13)	26 (0.27)

Solution Ⓐ > Solution Ⓑ >>> Solution Ⓒ

Key Findings

Model Size Matters

GPT-4 and CoPilot achieve the best performance in successfully pirating netlists

CodeLlama-13B performs significantly better than the smaller CodeLlama-7B

Key Findings

Model Size Matters

GPT-4 and CoPilot achieve the best performance in successfully pirating netlists

CodeLlama-13B performs significantly better than the smaller CodeLlama-7B

Training Data Size Matters

Latest version of Llama (Llama3-8B) outperforms the older Llama2 models

Key Findings

Model Size Matters

GPT-4 and CoPilot achieve the best performance in successfully pirating netlists

CodeLlama-13B performs significantly better than the smaller CodeLlama-7B

Training Data Size Matters

Latest version of Llama (Llama3-8B) outperforms the older Llama2 models

Feedback Improves Performance

With proper feedback and multiple attempts, smaller LLMs correct their mistakes

Thank You

Vasudev Gohil

vasudevgohil.com

References

[1] Yasaei, Rozhin, Shih-Yuan Yu, Emad Kasaeyan Naeini, and Mohammad Abdullah Al Faruque. “GNN4IP: Graph neural network for hardware intellectual property piracy detection.” In 2021 58th ACM/IEEE Design Automation Conference (DAC), pp. 217-222. IEEE, 2021.

[2] Aiken, Alex. “A System for Detecting Software Similarity.” https://theory.stanford.edu/~aiken/moss/

[3] Jplag. “JPlag - Detecting Software Plagiarism.” https://github.com/jplag/JPlag

[4] Grune, Dick. “The software and text similarity tester SIM.” https://dickgrune.com/Programs/similarity_tester/